Ripplation

株式会社リプレーション

Amazon Kinesisを用いたリアルタイムのゲーム分析システムと、AWS ElasticBeanstalk＆Amazon DynamoDB を活用したゲーム運営

株式会社リプレーション

モバイル系端末を中心としてゲーム開発を行っている。

コンシューマーの開発経験者が多数占めている。

自社タイトル「騎士とドラゴン」を運営中。


騎士とドラゴン
The Knight & The Dragon


準備
クエスト選択
バトル
ランク 8
ドラゴンハート
2
27825
そうび
28/30
そうび編成
そうび発却
強化合成
進化合成
大切登録
おしごと
クエスト
そうび
フレンド
ショップ
お宝箱
その他
グリニアの森
イベント島
200 / 200
HP
AP

1 画面下のスキルボタンをタッチして
敵を攻撃するだけ!!!

タッチだけの簡単バトル!

2 武器ごとにスキルが異なる!
自分だけの武器を見つけ出し、
使いこなせ! スキルの組み合わせ
次第で必殺技も!

スキルを使いこなせ!

3 敵の攻撃にあわせてガードを
すると、ダメージを防げるぞ!
タイミングが良いとすごいガードに!

タイミングよくガード!

「騎士とドラゴン」を作る上でのサーバー指針

- 如何にして最少人数でサーバー側の開発を行うか。
- 如何にしてサーバー運用コストを下げるか。

上記が達成できるならどんな環境・構成でも良いと割り切る！

試すことにしたのがAWS

一番気をつけたことは？

中途半端に使うのではなく、AWSを最大限効率よく利用する！

→環境依存とか気にしない！

どうしたか？

- 出来るだけ簡単に構築。
- 出来るだけ簡単にスケールや耐障害性の自動化。
- 監視システムも出来る限り簡単に。

AWS Elastic Beanstalk + Amazon DynamoDB
のみで出来る限り構成することにした！

AWSを使って見てわかったこと

- AWS触っていない場合でも直ぐ始められる。
- アプリをサーバーにデプロイするまでの流れを作るのが簡単！
  ※1から整える必要が無い。
- 思った以上に実用に耐えられる。
  →実際耐えている！
- ほとんど放置が可能。
  →最近サーバーをコンソール上で少し調整したことしかない。

AWS Elastic Beanstalk の注意点

- デプロイ時に一瞬アクセスできないことがある
- オートスケーリングの調整はきっちりしておくべし

DynamoDB の注意点

- 社内にいるDB技術者の文句やいちゃもんに負けないこと
- バッチ処理をしっかり使う。リトライも行う
- スループットの限界値は急激に変化させられないので注意
  ※急激な書き込みや読み込みが出来ない
- バックアップするかどうか
  ※リストアとかする状況ってどんな状況だろうか

汎用化なんて考えたら駄目

- AWSを使い倒さないと真の良さは発揮されない！

  PostgreSQL？MongoDB？**自前のロードバランサ？**

そんなにどれでも動く仕組みって重要なんでしょうか？

ログデータの収集・集計はどうする？

- 行動ログだけはRDSを使ったりすることも多いが、
  それももっと簡略化したかった。
  結局、最初はDynamoDBだけで何とかした。

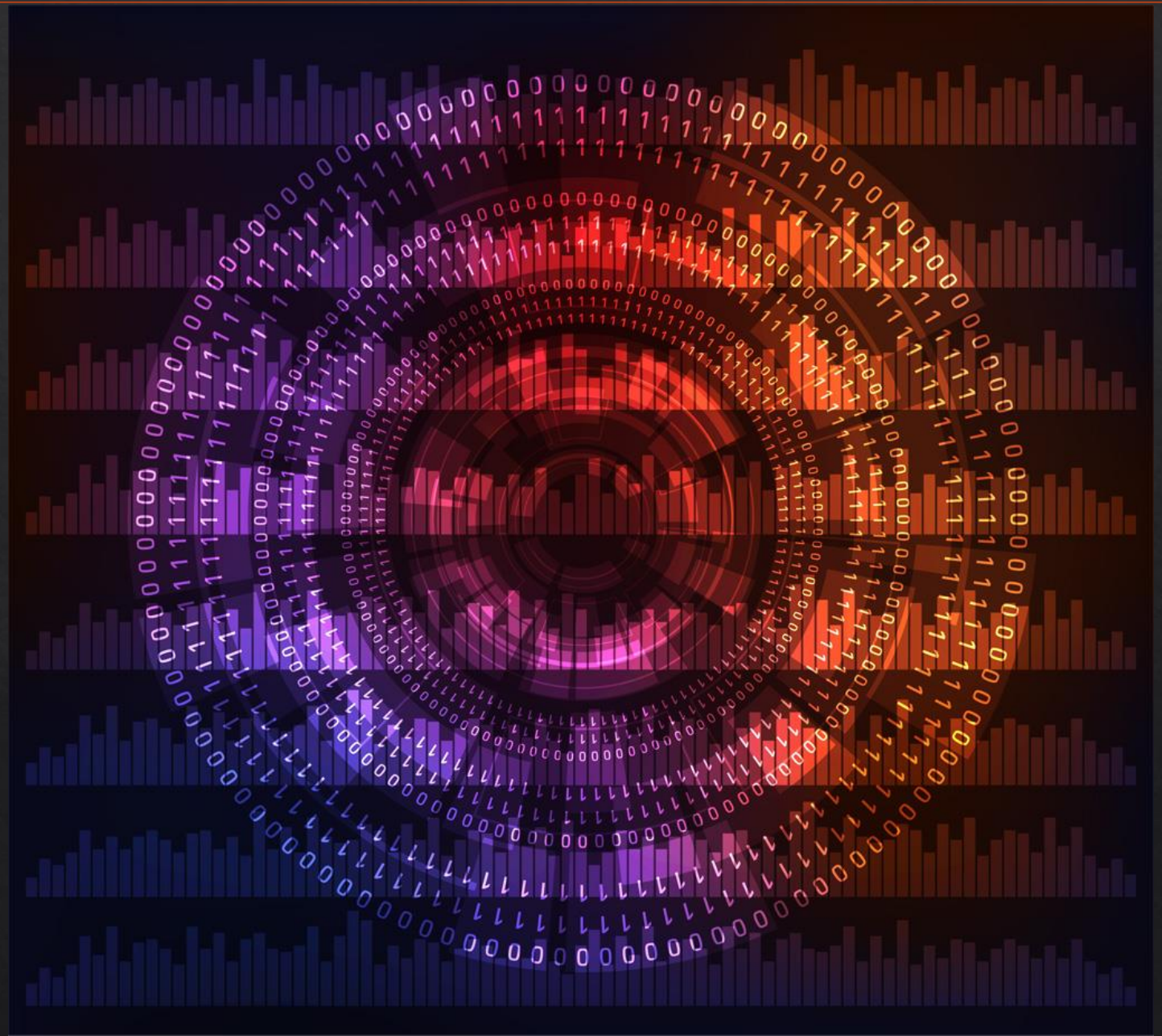
でもより簡単に、そしてより安くしたい。
では、自分たちでサービスにしてしまえ！！

KinesisとBig Data

～Kinesisが作る世界～

ビッグデータとは何か

Ripplation



©VectorBg.Net

現在のサーバの限界（※Linuxの場合）

Ripplation

▶ **達成課題**
- **秒間10万件のデータを収集する**

▶ 条件（デフォルト）
- Linux
- TCP通信によるデータ収集
- 割り当て可能ポート範囲：
  32768～61000（28,232個）
- TIME_WAIT：60秒
- 実測値への補正係数：0.6

28,232 / 60 * 0.6 ≒ 282（秒間処理可能数）

▶ 条件（チューニング後）
- Linux
- TCP通信によるデータ収集
- 割り当て可能ポート範囲：
  1024～65535（64,511個）
- TIME_WAIT：10秒
- 実測値への補正係数：0.6

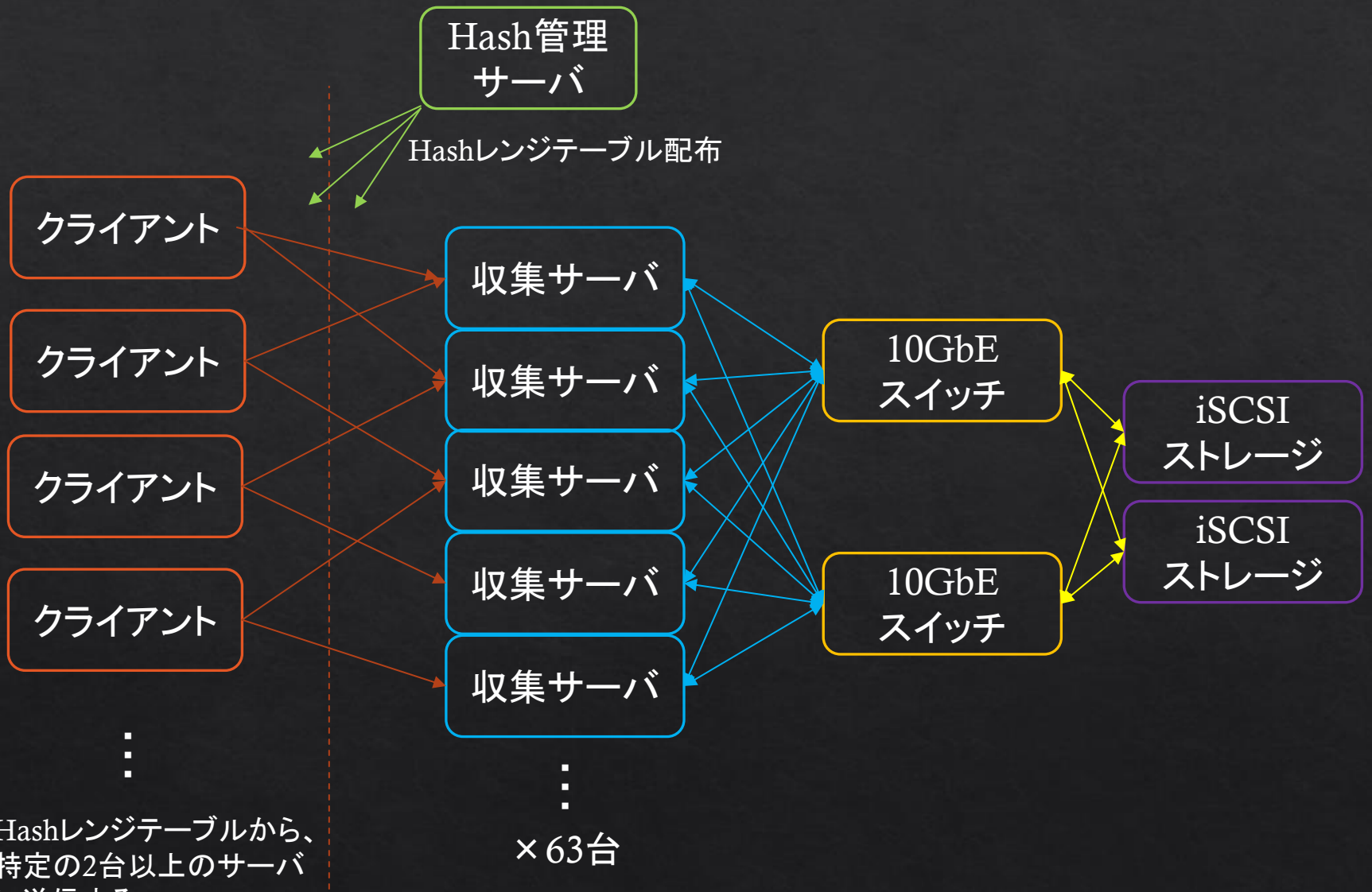
64,511 / 10 * 0.6 ≒ 3870（秒間処理可能数）

▶ 追加条件
- サーバ台数：26台

3,870 * 26 = 100,620（秒間処理可能数）

▶ 追加課題
- 26台への振り分け（バランシング）
- 収集データの保存

実際に必要となるサーバ構成



※1、冗長化用に2倍のデータを受付ける ⇒2倍
※2、iSCSIストレージにデータを送る⇒1.2倍

その他必要になりそうな事

・Hashレンジ管理用のプログラム
・Hashレンジ配布サーバの冗長化
・Hashレンジ関連サーバの障害時運用ケース作成
・Hashレンジ関連サーバの障害時復旧システム及びマニュアル
・収集サーバのスケールアウト関連システム
・収集サーバの障害時運用ケース作成
・収集サーバの障害時復旧システム及びマニュアル
・スイッチ障害時運用ケース作成
・スイッチ障害時復旧システム及びマニュアル
・ストレージ障害時運用ケース作成
・ストレージ障害時復旧システム及びマニュアル
・ストレージのDRシステム
・ストレージDRからの復旧システム及びマニュアル
・各種サーバ監視システム（SNMP的なもので）
・監視システムからのアラート送信、受信システム
・障害時対応用オペレータ
・サーバセンターでのシステム構築を前提としたSI

⋮

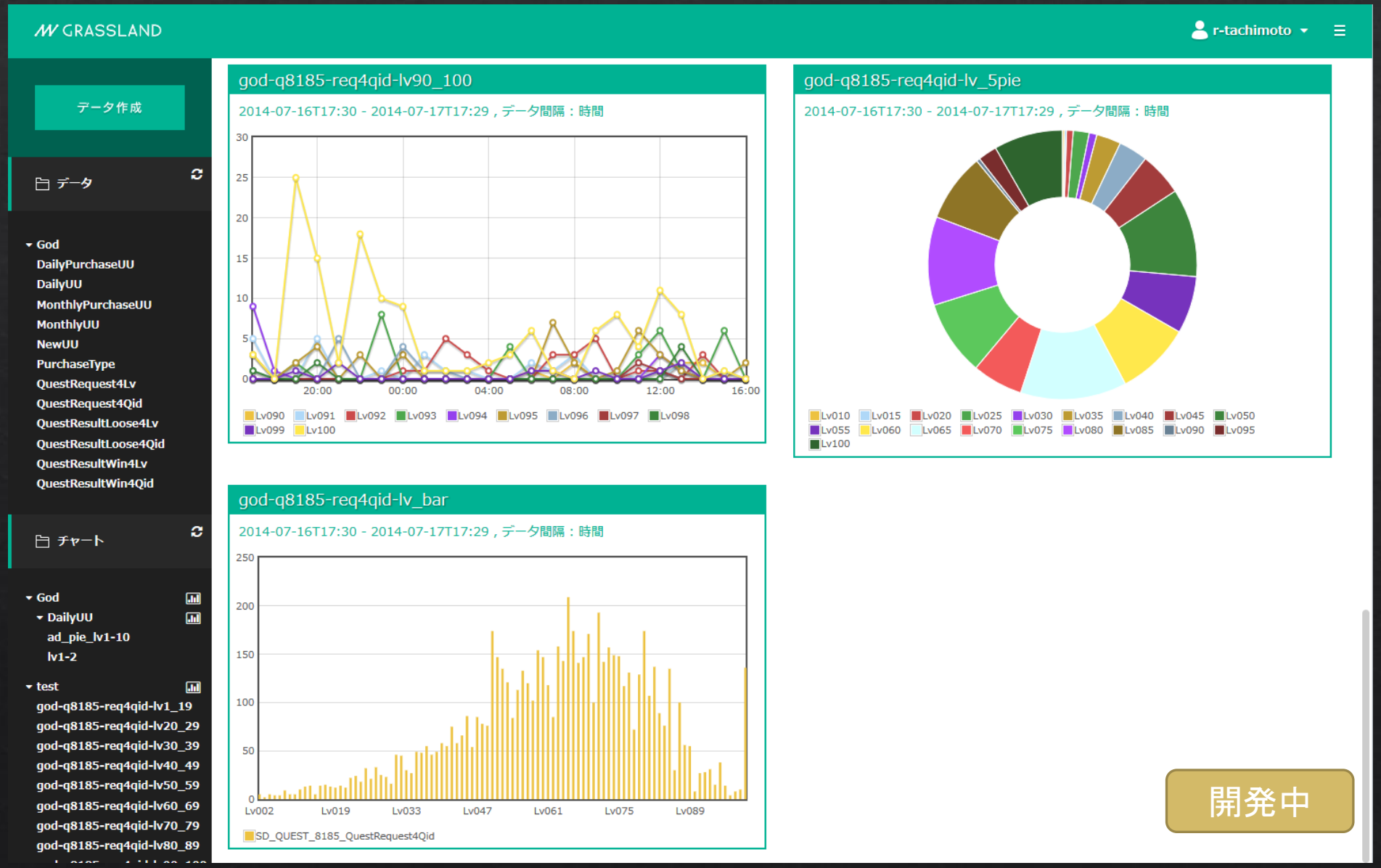
お金も期間もそれなりに・・・

Amazon Kinesis

×

100 Shards

Grassland紹介



Grasslandスペック

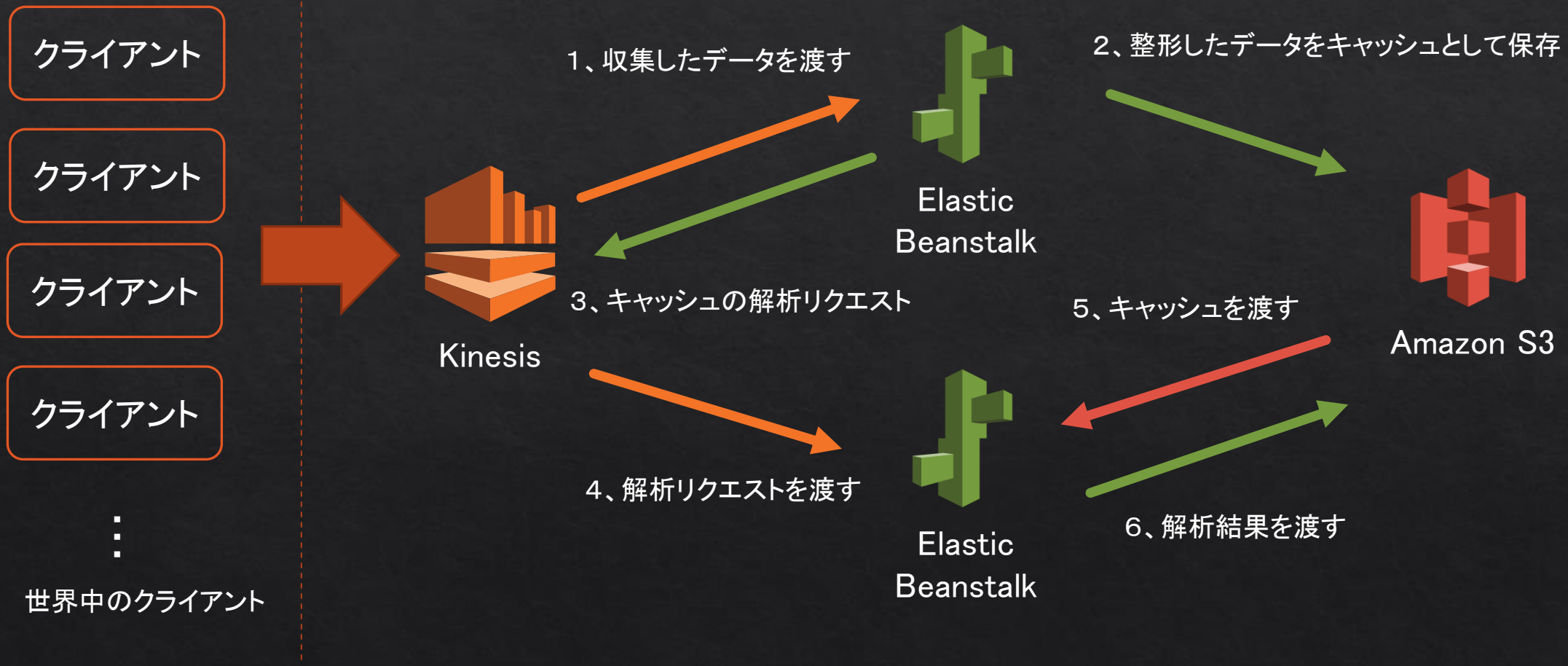
- なんでも集計できます
- なんでも解析できます
- 毎時180億レコードまで捌けます

世界中の全ての集計を一手に担える規模設計です

GrasslandでのKinesisの使いどころ

Ripplation

▸ Grasslandの収集、解析フロー



Kinesisが作る世界

Ripplation

©guzzphoto

有難う御座いました

KinesisとBig Data ～Kinesisが作る世界～